

# FarmFantasy

30 Août 2016

Ramushi Ardi  
Paschoud Nicolas



<b>Description</b>	<b>3</b>
<b>Caractéristiques techniques</b>	<b>3</b>
Visual Studio 2013	3
C#	3
<b>Fonctionnalités du jeux</b>	<b>3</b>
Gestion de l'argent	3
Magasin	3
Sauvegarde	3
Plusieurs variétés de semences	3
4 Types de bétails	3
<b>Diagramme de classe</b>	<b>4</b>
<b>MCD</b>	<b>6</b>
<b>Maquettes</b>	<b>7</b>
Fenêtre de login (frmLogin) :	7
Fenêtre principale (frmMain) :	7
Fenêtre magasin (frmMagasin)	8
Fenêtre de l'entrepôt :	8
<b>Cas d'utilisations</b>	<b>9</b>
<b>Variables</b>	<b>11</b>
<b>Pseudo code</b>	<b>12</b>
<b>Améliorations futures</b>	<b>15</b>
<b>Conclusion</b>	<b>15</b>

## Description

FarmFantasy est un jeu de gestion de ferme évolutif. Plusieurs animaux et type de cultures seront disponible ainsi que des améliorations de bâtiments.

Avant de lancer le jeux, il faut importer la base de données (se trouvant dans les fichiers source), sur un serveur local

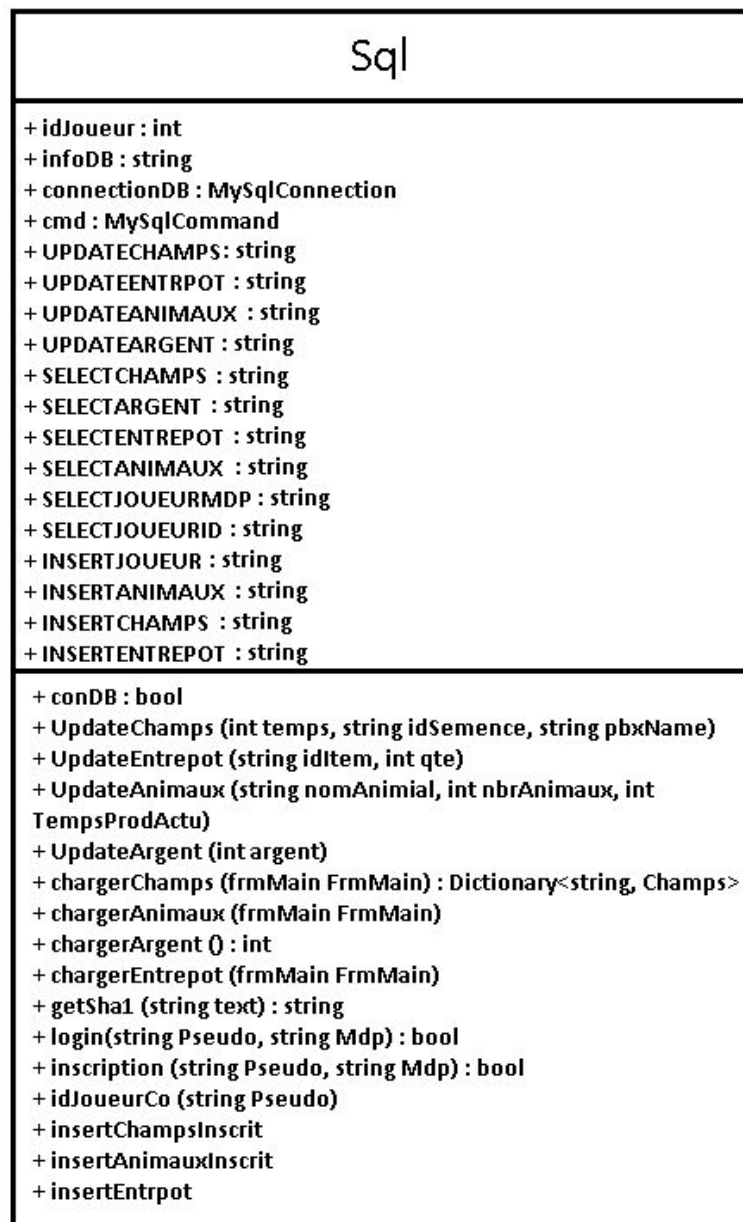
## Caractéristiques techniques

- Visual Studio 2013
  - C#
    - Windows Form
    - Objets
    - Ressources
    - Base de données
    - Interface multiforme
    - Fenêtre à propos

## Fonctionnalités du jeux

- Gestion de l'argent
  - Magasin
    - Vente de semences
    - Vente de produits
    - Achat de semences (Blé, Maïs, Colza,...)
    - Achat de bétails
    - Achat de produits
- Sauvegarde
  - Sauvegarde à la fermeture
  - Possibilité de recommencer la partie
- Plusieurs variétés de semences
  - Blé
  - Carotte
  - Patate
  - Maïs
  - Colza
- 4 Types de bétails
  - Vaches
  - Poules
  - Cochons
  - Moutons
- Jouer à la souris

## Diagramme de classe



La classe Sql, contient toutes les requête nécessaire :

- à l'inscription d'un joueur
- à l'enregistrement d'une sauvegarde
- au chargement d'une sauvegarde
- à la connexion d'un joueur

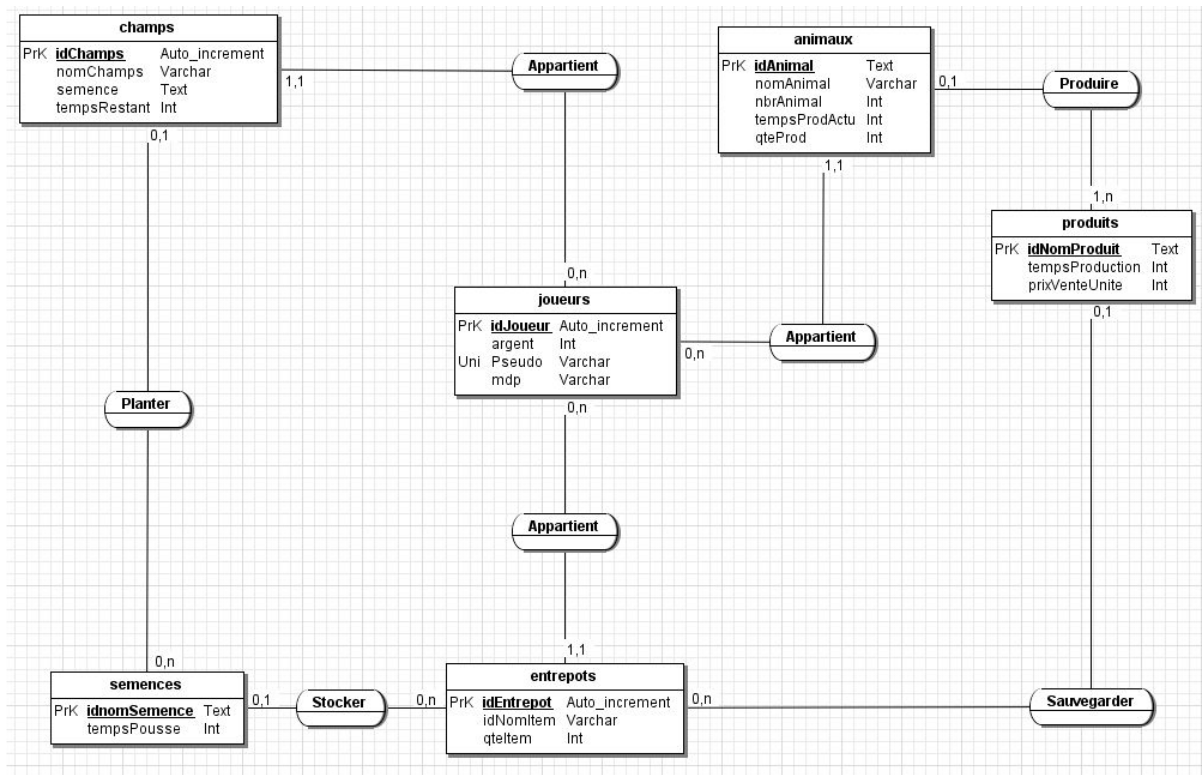
Animaux
- _typeAnimal : string - _nbrAnimaux : int - _tempsProduction : double - _prixVenteTot : double - _temps = 0 : int - _prixV : double - _quantite : int
+ <<prop>> PrixV : double + <<prop>> TypeAnimal : string + <<prop>> TempsProduction : double + <<prop>> Temps : int + <<prop>> Qunatite : int + <<prop>> NbrAnimaux : int + <<prop>> PrixVenteTot : double
+ <<ctor>> Animaux(double prixV, double tempsProd, string typAnim, int qteProd) + majPrix () + calculTempsProd () : bool

La classe Animaux, contient tous les paramètres utile à la production d'un animal

Champs
- _tempsPousse : double - _pbxChamps : PictureBox - _culture : string - _temps = 0: int - DicoSemence : Dictionary<string, double>
+ <<prop>> Temps : int + <<prop>> PbxChamps : PictureBox + <<prop>> Culture : string + <<prop>> TempsPousse : double
+ <<ctr>> Champs (PictureBox champs, string semence) + <<ctor>> Champs (PictureBox champs, string semence, int TempsActu) + calculTemps() : bool

La classe Champs, contient tous les paramètres utile à la récolte d'un champ

## MCD



## Maquettes

Fenêtre de login (frmLogin) :

Maquette de la fenêtre de login (frmLogin). Le titre de la fenêtre est "Titre du jeu". À l'intérieur, il y a trois paires de "Label" et "TextBox" alignées horizontalement. En bas à droite, il y a deux boutons : "Btn Connexion" et "Btn Inscription".

Fenêtre principale (frmMain) :

Maquette de la fenêtre principale (frmMain). La fenêtre contient plusieurs éléments :  
- En haut à gauche : un bouton "bouton".  
- En haut à droite : deux labels "label" et un bouton "BtnSauvegarden".  
- À gauche : une zone "PictureBox des champs" contenant un rectangle "Entrepot".  
- À droite : une section "Semences" avec un bouton "Planter" et un "RadioBouton".  
- En dessous de "Semences" : un bouton "Bouton Magasin".  
- En bas à droite : une section "Animaux" contenant deux colonnes de labels "label" sous les titres "Animaux" et "Entrepot".

## Fenêtre magasin (frmMagasin)

The diagram illustrates the layout of the 'Fenêtre magasin (frmMagasin)' window. It features a main container with a 'Label' at the top left. Below this, there are two side-by-side panels:

- Semence Panel:**
  - Contains a 'ComboBox' and a 'NumericUpDown' control.
  - Below these are three stacked 'Label' controls.
  - At the bottom, there are two buttons: 'Bouton vendre' and 'Bouton Acheter'.
- Animaux Panel:**
  - Contains a 'ComboBox' and a 'NumericUpDown' control.
  - Below these are three stacked 'Label' controls.
  - At the bottom, there is a single button: 'bouton Acheter'.

## Fenêtre de l'entrepôt :

The diagram illustrates the layout of the 'Fenêtre de l'entrepôt' window. It consists of a single, large 'ListBox' control within a rectangular frame.



## Cas d'utilisations

Nom	Sauvegarder le jeu
Description	L'utilisateur clique sur le bouton sauvegarder qui se trouve sur la page principale
Acteurs	Le programme et l'utilisateur (page frmMain)
Résultat	Le jeu se sauvegarde dans la base de données
Séquence d'événements	
Quand l'utilisateur souhaite se souvenir de l'état actuel de sa partie	
Exception : Connection impossible	
Le programme ne peut pas se connecter à la base de données La partie n'est pas sauvegardés	

Nom	Charger une partie
Description	La partie de l'utilisateur est chargée après la connexion au jeu
Acteurs	Le programme et l'utilisateur (page frmMain)
Résultat	Le programme doit avoir chargé la dernière sauvegarde de l'utilisateur
Séquence d'événements	
Quand l'utilisateur a cliqué sur le bouton Login de la page frmLogin	
Exception : Connection impossible	
Le login est erroné Le mot de passe est erroné La connection à la base de données est impossible	

Nom	Accéder à la fenêtre magasin
Description	L'utilisateur souhaite accéder au magasin pour vendre / acheter des produits / animaux
Acteurs	L'utilisateur et le programme (frmMain et frmMagasin)
Résultat	La fenêtre du magasin doit s'ouvrir et récupérer toutes les données des produits (quantité) et animaux (possession) ainsi que l'argent actuel
Séquence d'événements	
L'utilisateur clique sur le bouton "Magasin" sur la fenêtre principale	

<b>Nom</b>	<b>Ouvrir l'entrepôt</b>
<b>Description</b>	L'utilisateur clique sur l'entrepôt qui se trouve à gauche, sur la fenêtre principale, afin de voir les produits possédés
<b>Acteurs</b>	L'utilisateur et l'application (frmMain et frmEntrepot)
<b>Résultat</b>	La fenêtre de l'entrepôt doit s'ouvrir avec les informations du nombre de produits possédés
<b>Séquence d'événements</b>	
Clique sur la pictureBox de l'entrepôt	

<b>Nom</b>	<b>Acheter des produits</b>
<b>Description</b>	L'utilisateur veut acheter des produits et va dans la fenêtre magasin pour faire ses achats.
<b>Acteurs</b>	L'utilisateur et l'application (frmMagasin)
<b>Résultat</b>	L'utilisateur a plus de produits
<b>Séquence d'événements</b>	
Sélectionne un produit avec le comboBox et choisit la quantité avec le numeriqueUpDown et clique sur le bouton Acheter de la fenêtre.	
<b>Exception</b>	
L'utilisateur n'a pas assez d'argent.	

## Variables

Login de type string

Sert à récupérer le login de l'utilisateur.

Entrepôt est un dictionnaire qui prend un string en index et prend comme valeur un int.

Un dictionnaire de données pour l'entrepôt.

repertoryAnimaux est un dictionnaire qui prend un string en index et prend Animaux comme valeur.

Un dictionnaire qui contient les objets pour chaque type d'animal

repertoryChamps est un dictionnaire qui prend un string en index et prend Champs comme valeur.

Un dictionnaire qui contient les objets pour chaque champs

FrmMagasin est une variable de type frmMagasin

Instanciation de la fenêtre magasin.

NBRCHAMPS est une constante

Cette constante indique le nombre de champs dans le jeu.

argent est une variable de type int et prend 100 comme valeur de base

L'argent de l'utilisateur.

## Pseudo code

### Chaque tick du timer (1 tick = 1 seconde)

Ce code permet de savoir si un champ a fini de pousser ou non

si le tableau de champs n'est pas vide

début si

pour i allant de 1 aux NOMBRE DE CHAMPS, incrémenter i

début pour

si l'index "pbxChamps" + i est dans le tableau

début si

alors on récupère l'objet qui se trouve à l'index ["pbxChamps" + i] en objet de type Champs dans le dictionnaire repertoryChamps

si la fonction calculTemps de nativChamps retourne vrai

alors on ajoute 2 de la semence de la propriété culture de l'objet nativeChamps dans l'entrepot

on supprime l'index "pbxChamps" + i, dans le Dictionnaire repertoryChamps

fin si

fin si

fin pour

fin si

### Chaque tick du timer (1 tick = 1 seconde)

Ce code permet de savoir si un animal a produit quelque chose ou non

si la propriété NbrAnimaux de l'objet Animal qui se trouve à l'index "vache" du dictionnaire repertoryAnimaux est plus grand que 0

début si

si la methode calculerTempProd de l'objet qui se trouve à l'index "vache" du dictionnaire repertoryAnimaux retourne vrai

début si

on incrémente la valeur se trouvant à entrepot["lait"] de la propriété Quantite de l'objet qui se trouve à l'index "vache" du dictionnaire repertoryAnimaux

fin si

fin si

### **Méthode calculTemps de la classe Champs**

Cette méthode détermine si la semence a fini de pousser. C'est une méthode de type booléenne

fini est un booléen initialisé à faux

La propriété Temps est incrémenté de 1

Si la propriété Temps est supérieur ou égal à la propriété TempsPousse

Début si

fini prend vrai

La propriété Enabled de la propriété PbxChamps prend vrai

La propriété Image de la propriété PbxChamps prend Properties.Resources.terre

La propriété Temps prend 0

fin si

La méthode retourne fini

### **Méthode calculTempsProd de la classe Animaux**

Cette méthode détermine si l'animal a fini de produire. C'est une méthode de type booléenne

fini est de type booléen et est initialisé à faux

Incrémenter Temps de 1

Si la propriété Temps est supérieur ou égal à la propriété TempsProduction

Début si

fini prend vrai

Temps prend 0

Fin si

Retourner fini

### **Méthode pbxClickChamps\_Click de frmMain**

Cette méthode permet de planter une semence sur n'importe quel des champs du jeu

pbx est une variable de type PictureBox qui prend le paramètre sender en tant que variable PictureBox

si le radioButton "rnbNothing" n'est pas checkée

alors début si

culture est une variable de type string qui prend string.Empty

si le radioButton "rnbBle" est checkée

alors début si

clture prend "ble"

sinon si le radioButton "rnbColza" est checkée

alors début si

clture prend "colza"

sinon si le radioButton "rnbCarotte" est checkée

alors début si

clture prend "carotte"

sinon si le radioButton "rnbPatate" est checkée

```
    alors début si
    clture prend "patate"
sinon si le radioButton "rbnMais" est checkée
    alors début si
    clture prend "mais"

si la quantité a l'index culture du dictionnaire entrepot est supérieur a 0
    alors début si
    l'index culture du dictionnaire entrepot est décrémenté
    la propriété Enabled de pbx prend false
    la propriété Image de pbx prend
    Properties.Resources.ResourceManager.GetObject(culture) caster en type
    Image
    Appeler la méthode majInterface()
    si repertoryChamps ne contient pas la clé au nom de la propriété Name de
    pbx
    Alors début si
    On ajoute un nouvel élément dans repertoryChamps avec comme index la
    propriété name de pbx, et un nouvel objet de type Champs qui prend en
    paramètre pbx et culture
```

## Améliorations futures

Meilleurs gestion de la connexion à la base de données

## Conclusion

Le programme est bien abouti, c'est un projet que nous avons tenté de réaliser en première année de notre formation. Il a été assez vite abandonné par manque de connaissance.